

**Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский государственный технический университет имени Н.Э. Баумана**

**(национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)**

**ОТЧЕТ**

По лабораторной работе №2

По курсу «Методы вычислений»

Тема: «Метод золотого сечения»

Вариант 2

Студент:

Апальков Ф.С.

Группа:

ИУ7-13М

Преподаватель:

Власов П. А.

Москва 2022

**ЛАБОРАТОРНАЯ РАБОТА№ 2**

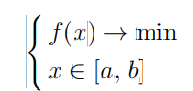
Метод золотого сечения

Цель работы: изучение метода золотого сечения для решения задачи одномерной минимизации.

Содержание работы

1. реализовать метод поразрядного поиска в виде программы на ЭВМ1 ;

2. провести решение задачи



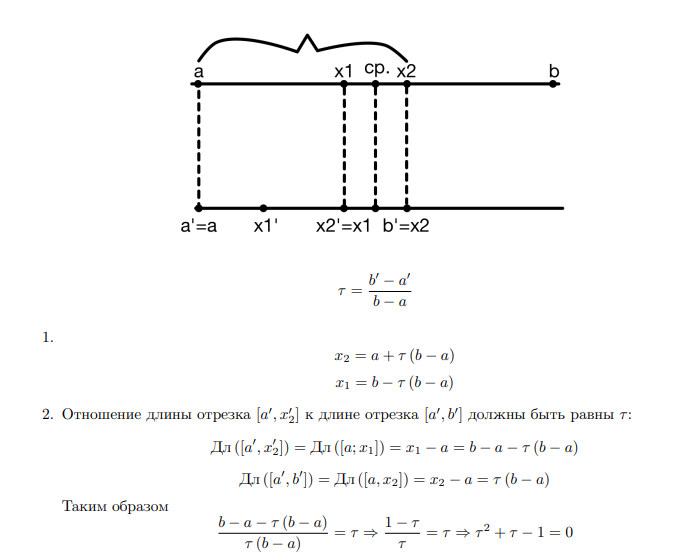
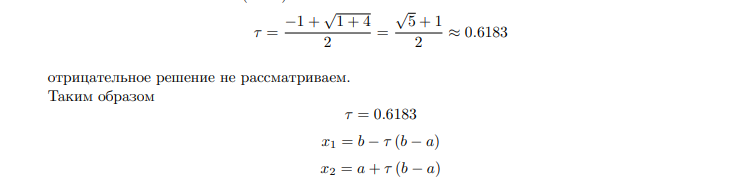
3. организовать вывод на экран графика целевой функции, найденной точки минимума (x∗, f(x∗)) и последовательности точек (xi, f(xi)), приближающих точку искомого минимума (для последовательности точек следует предусмотреть возможность” отключения” вывода ее на экран

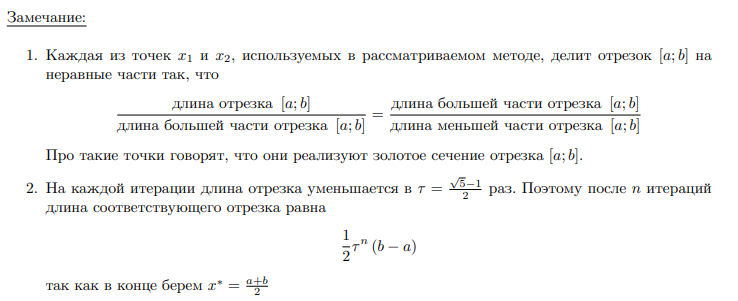
Данные индивидуального варианта:

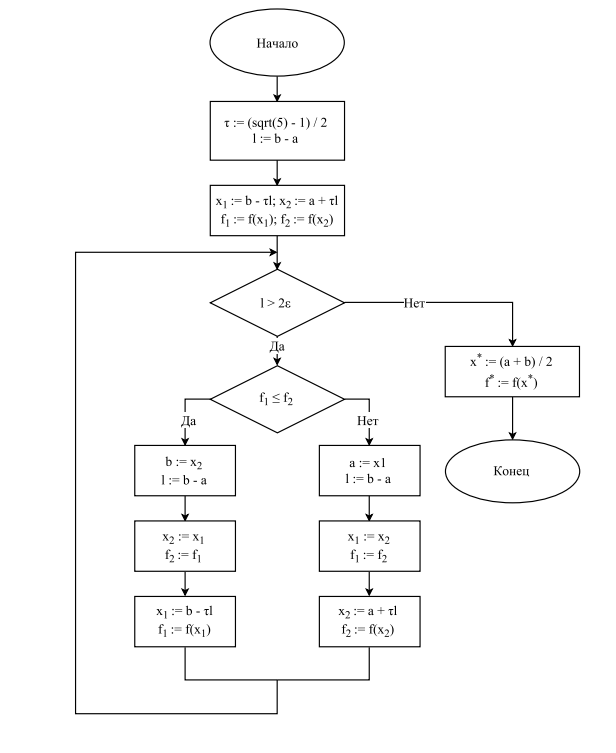
Функция [a,b]



**Теоретическая часть**

Для уменьшения количества значений целевой функции, которые приходится вычислять, постараемся выбрать пробные точки x1 и x2 внутри отрезка [a; b] так, чтобы при переходе к очередному отрезку одна из этих точек стала новой пробной точкой. При этом будем считать, что отношение длины нового отрезка к длине текущего отрезка не зависит от номера итерации и равно τ. Так же будем считать, что x1 и x2 располагаются симметрично относительно середины отрезка [a; b].





**Код программы**

function lab2()

FX = @(x) (cos(power(x,5) - x + 3 + power(2, 1/3)) + atan( (power(x,3) - 5 \* sqrt(2)\*x - 4) / (sqrt(6)\*x + sqrt(x)) ) + 1.8);

clc();

a = 0;

b = 1;

start\_a = a;

start\_b = b;

epsilon = 0.01;

plot\_a\_x = [];

plot\_a\_y = [];

plot\_b\_x = [];

plot\_b\_y = [];

tau = (sqrt(5) - 1) / 2;

l = b - a;

x1 = b - tau \* l;

x2 = a + tau \* l;

f1 = FX(x1);

f2 = FX(x2);

N = 2;

while 1

if l > 2 \* epsilon

if f1 <= f2

b = x2;

l = b - a;

x2 = x1;

f2 = f1;

x1 = b - tau \* l;

f1 = FX(x1);

else

a = x1;

l = b - a;

x1 = x2;

f1 = f2;

x2 = a + tau \* l;

f2 = FX(x2);

end

N = N + 1;

plot\_a\_x(end+1) = a;

plot\_a\_y(end+1) = FX(a);

plot\_b\_x(end+1) = b;

plot\_b\_y(end+1) = FX(b);

else

break

end

end

res\_x = (a + b) / 2;

res\_fx= FX(res\_x);

fprintf('N = %d e = %f x\* = %f f(x\*) = %f\n\n\n', N, epsilon, res\_x, res\_fx);

fplot(FX, [start\_a, start\_b]);

hold on;

scatter(res\_x, res\_fx);

hold on;

plot(plot\_a\_x, plot\_a\_y, '\*', plot\_b\_x, plot\_b\_y, '\*');

**Результаты расчётов**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п\п** | **ε** | **N** | **x\*** | **f(x\*)** |
| 1 | 0.01 | 12 | 0.611456 | -0.325921 |
| 2 | 0.0001 | 21 | 0.614002 | -0.325929 |
| 2 | 0.000001 | 31 | 0.614038 | -0.325929 |

